Volume 2, No. 2, Mar-Apr 2011
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

A Study on Context–based Retrieval Trends and Techniques

Dr.M.Thangaraj
Associate Professor, Dept. of Computer Science,
M.K. University, Madurai,
TamilNadu, INDIA.
thangarajmku@yahoo.com

V.Gayathri*
Research Scholar, Dept. of Computer Science,
M.K. University, Madurai,
TamilNadu, INDIA.
gayathrivengatmku@yahoo.com

Abstract: With the tremendous growth of the Web, information "Big Bang" has been taken place on the Internet. Search Engines have become one of the most helpful tools for obtaining useful information from the "Big Bang". Increasing growth of information volume in the Internet causes an increasing need to develop automatic methods for retrieval of documents and ranking them according to their relevance to the query. Although various methods for the efficient retrieval of information were presented, still the end users are struggling to get the desired result. At present the major issues in searching are (a) topic diffusion: results returned by a keyword based search, fall into multiple topic areas, which are not interested to users; (b) there is no effective scoring mechanism; so the users are forced to scan a large result set, which leads them to miss the important ones. To resolve these issues context – based retrieval is used. This paper explores the research trends and techniques by reviewing various approaches and lists the research issues in this field.

Keywords: context-based retrieval; information retrieval; ontology-based search; relation-based search; similarity

I. INTRODUCTION

The web is rapidly growing and becoming a huge repository of information, with several billion pages and more than 300 million of users globally [15]. This information volume causes many problems that relate to difficulty of finding, organizing, accessing, and maintaining the required information by users. Correspondingly, extensive methods and algorithms to reduce or rank this digital information are being researched to assist users in their searches.

When searching for digital information, users enter keyword search terms to a query interface or search engine. To reduce the user time in examining results, the returned documents, which relate to the keyword search term, are sorted according to their popularity. When a keyword-based search returns documents from a diverse set of topics, popularity ordered results might not be beneficial to all users.

To resolve the above-summarized problem, the notion of context-based searching is used. Defining a search context allows us to delineate a smaller and less diverse set of searched documents. By guiding users to a search context beforehand, we can retrieve documents that are more closely associated in meaning with the search terms. For users who are unable to exactly state the correct search terms, defining a context increases the odds of quickly finding relevant documents. For search terms returning an assortment of unrelated topics, context-based search reduces topic diversity and returns a more meaningful document rank order.

II. OVERVIEW

The directions of research in a context – based search were classified into i) Context Searching and Categorization, ii) Searching based on Similarity, iii) Ranking, and other research issues (Fig. 1).

In the first phase, the need for classifying the corpus is to be decided. The role of second phase is to search with similarity and identifying the relevant documents. Ranking of relevant documents are carried out in the third phase.

The remainder of the paper is organized as follows: Section III is devoted to the issues relevant to searching with categorization. In Section IV, we discuss searching based on similarity. Section V deals with ranking. And section VI describes other research issues in this field. Finally, in Section VII we present conclusion and directions for future research in this area.

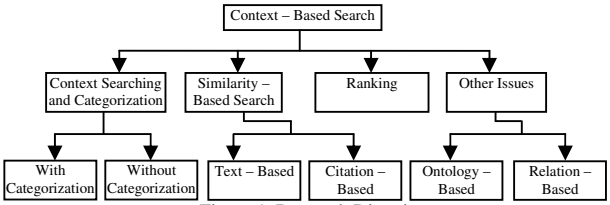

Figure 1. Research Directions.

III. CONTEXT SEARCH AND CATEGORIZATION

In many existing systems, contexts (or concepts) have been used to reduce the topic diffusion problem among documents. Different systems define the notion of context in different ways.

A. *Context Search with No Categorization*

TileBars [13], lets the user enter a query in a faceted format (i.e., each line represents each topic) and provides graphical bar in order to show the degree of match for each facet. TileBars illustrate which parts of each document contain topic by dividing the bar into columns, where each column refers to a part in the document. The darkness of the square indicates the number of times the topic occurs in the part of the document. With this approach, the user can easily see the relevancy of the document to each specified topics. However, search results are shown as single list and no categorization of search results is provided.

Similarly contextual web search approach, e.g. Y!Q Contextual Search [19] and IntelliZap [9], a context is captured around the user-highlighted text, and augmented queries are created from the selected context words. The users

can specify contexts of interests before viewing search results and no structural and hierarchical information are used. Sometimes user need not give keyword to initiate the search, eg. in [28], according to the environment variables, contexts are selected automatically. Results are retrieved for the set of predefined query based on the corresponding context. The user can select from the list of results that are generated automatically.

B. *Search Systems with Categorization*

There is a variety of categorization techniques have been proposed to make search results more understandable. Two broad categorization techniques are document clustering and document classification. Clustering creates categories (or contexts) by grouping similar documents together while classification assigns documents to a set of predefined categories.

1) Clustering – based Search

Scatter / Gather [14] was one of the first clustering systems on top of the Information Retrieval engine, in which it groups documents based on the similarities in their contents. Grouper [49] uses Suffix Tree Clustering (STC) that identifies sets of documents sharing common phrases. Lingo [24] uses singular value decomposition (SVD) to find meaningful labels for the clusters.

Findex [17] seeks for the most frequent words or phrases among search results and use them to define categories, which are displayed in a separate list beside the results. In [32], a clustering method with fuzzy logic is used. Initially, the system pre-processes the input document. Document property vector is obtained from the terms in the document. Based on the predetermined threshold value, right clusters for that document, can be found. Documents in this cluster are called as candidate documents.

Similarity is determined by comparing category property vector that includes degrees of documents belonging to clusters and input document category property vector. Distance based similarity measurement is defined for this comparison. A document which is at the process of training in cluster operation can belong to more than one category, but for the document under test stage is assumed that belong to only one cluster is a problem to be resolved. Final comparison operation is done only by documents in this determined cluster.

SemreX [12], a semantic overlay for desktop literature / document retrieval in peer-to-peer networks. Semantically similar peers are locally clustered together. Based on the semantic overlay, a heuristic query routing algorithm is proposed for efficient content searching. First, logical concept layer which aims at providing the shared concept model (ontology) over the distributed P2P network. Second, document object space layer, which is to classify a single document into Topics, LSI applied to reveal semantic subspaces of feature spaces from documents stored on peers. After producing semantic vectors through LSI, it is trained to classify the documents into different categories based on the extracted vectors, by training SVM.

SVM creates a hyperplan that separates the training data with the maximum-margin into a feature space induced by a kernel function used as the inner product. Third, semantic overlay layer, in which, each peer knows its local neighbors with similar interest at high probability approximating to p = 1; and, each node knows a small number of randomly chosen distant neighbors. Forth, the underlying peer communication layer, which serves as a transport layer for other layers of the system model and hides all low-level communication details from the rest of the layers. But, it is capable of handling only one keyword. Similarly in [10], with the user feedback information, training is given to the system using SVM, and the context preferences are extracted, based on which documents are stored. When a query is given, the documents are retrieved based on the corresponding context which is extracted from the clusters.

2) Classification – based Search

Several existing information retrieval systems utilize document classification techniques to improve the search experience. In [2], documents are classified based on the information collected from the background of user's working environment using naïve bayesian classifier. Essentially, every document pair form a subset of documents that relate to the same task is evaluated for a possible relationship. For each document pair a corresponding dataset is formed. From a combination of the user-provided data and the data captured automatically, training examples with a document relationship classification can be used to populate the training set. When a long session time used, then the context may vary. Similarly, if the documents of the same session are not relevant to each other, then there won't be any meaningful usage relationships.

Similarly in [6], the Bayes formula was used to vectorize (as opposed to classify) a document according to a probability distribution reflecting the probable categories of the document. The Bayes formula gives a range of probabilities to which the document can be assigned according to a predetermined set of topics (categories). Using this probability distribution as the vectors to represent the document, the SVM can then be used to classify the documents on a multidimensional level. If a single keyword represents multiple contexts, then this system will produce highly inaccurate results.

Mostly the vector space model (VSM) is used for vectorization [40], in which, each unique term in the vocabulary represents one dimension; thus the dimensions in feature space is increased. Latent semantic indexing (LSI) is a successful technology in information retrieval which attempts to explore the latent semantics implied by a query or a document through representing them in a dimension-reduced space.

IV. SIMILARITY BASED SEARCH

A number of existing works are proposed to measure the similarity of two objects by using relevant information of the two objects. Generally there are two types of approaches to calculate the similarity. They are *Text based*, in which similarity is calculated by the number of times the keywords in the query appearing in the document, and *Citation – based*, in which the count i.e., how many times the particular page is cited, is considered for calculating the similarity. The following section describes the research issues in these fields.

A. *Text – based Similarity*

There were several kinds of information retrieval (IR) models, such as the Boolean, the vector space, the probabilistic, the connectionistic, the cluster, the rule-based, the fuzzy logic or fuzzy set, the semantic model, etc. In general, an IR model specifies a document representation, a query representation and a matching procedure. The majority of IR system is based on the Boolean model; however, the vector space model is the most frequently used in experimental systems.

Generally IR systems work based on the frequency of a keyword appeared in documents. The weight by considering term frequency and inverse document frequency (*tf – idf*) was first studied in [34]. tf_{ij} is defined as the number of occurrences of keyword k_j in document di and idf_j defined as $log(N/df_j)$ in which N is the total number of documents and df_j is the number of documents containing keyword k_j. Based on this concept,

several researchers further modified the formula into various forms [46].

It is not language dependent. It is also applicable for other languages like Tamil, etc. In [30], Tamil text classification system based on vector space model and neural network model is presented. Once the document is represented as weight matrix, based on the *tf – idf*, we can apply any one of the distance measures such as Euclidean distance, Mahalanobis distance, Manhatan distance or cosine measure to find the similarity of documents.

Once the frequencies are calculated, then the similarity between two words or terms can be calculated by the cosine similarity as follows.

$$s^K(w_1, w_2) = \frac{\sum_{i=1}^{N} {}^t w_1, i {}^t w_2, i}{\sqrt{\sum_{i=1}^{N} ({}^t w_1, i)^2} \sqrt{\sum_{i=1}^{N} ({}^t w_2, i)^2}} \quad (1)$$

Large vector space is created when *tf – idf* is applied that leads to more computational overhead. The *tf – idf* is used, not only in the text – based approach; it is can also be applied in the graph – based representation. In [37], a document is represented by undirected and directed graph, respectively. Then terms and vectorized graph connectionists are extracted from the graphs by applying several feature extraction methods. A document retrieval system based on self-organizing map (SOM) is developed to speed up the retrieval process. In order to minimize time consuming matching process, first, we extract term-connections from graph representations with extensive feature extraction methods. Each document is then projected into feature vector space forming term-connection-frequency (*tcf*) together with term frequency (*tf*).

Term – Frequency: as in

$$w_t = \sqrt{f_t} \times idf \quad \text{where } idf = \log_2\left(\frac{N}{f_d^t}\right) \quad (2)$$

Term – Connection – Frequency:

Three schemes are used to extract the term-connection-frequency to construct a term-connection based vocabulary. The first one, is top term-connection-based method that is to select the most frequent N_{tc} term-connections from matrix A, where $A_{ij}^K = f_{ij}^{k,tc}$. Second, we use the same weighting measure to calculate the weight of each term-connection for a pair of terms

$$w_{ij}^{tc} = \sqrt{f_{ij}^{tc}} \times idf_{ij}^{tc} \text{ where } idf_{ij}^{tc} = \log_2\left(\frac{N}{f_{d,ij}^{tc}}\right) \quad (3)$$

Then, we sort the term-connections by using the weights in descending order and select the first N_{tc} term-connections. Finally, we use a similar entropy-based measure to weight each term-connection

$$entropy_{ij} = \sum_{k=1}^{N} \frac{A_{ij}^k}{f_{d,ij}^{tc}} \log_2\left(\frac{f_{d,ij}^{tc}}{A_{ij}^k}\right) \quad (4)$$

Sometimes, frequency can be calculated with category. In [44], the *TF-IDF* formula can be intuitively extended into the *TF-ICF* (term frequency-inverse category frequency) formula to represent a category with a vector of terms. Thus it is needed more research in this field to perform effectively in terms of scalability, and accuracy.

B. Citation – based Similarity

To present the documents in an ordered manner, Page ranking methods are applied, which can arrange the documents, in the order of their relevance, importance, and content score and web mining techniques are also used.

Surgey Brin and Larry Page [27] developed a ranking algorithm used by Google, named PageRank (PR) after Larry Page (cofounder of Google search engine), that uses the link structure of the web to determine the importance of web pages. Google uses Page Rank to order its search results so that documents that are seem more important move up in the results of a search accordingly.

This algorithm states that, if a page has some important incoming links to it, then its outgoing links to other pages also become important. Therefore, it takes back links into account and propagates the ranking through links. Thus, a page obtains a high rank if the sum of the ranks of its back links is high. Later various versions of the Page Rank are derived [23], which is presented in Table. I.

Table I. Various Page Rank Algorithms

Scheme	*Equation*	*Description*
Simplified version	$PR(u) = c \sum_{v \in B(u)} \frac{PR(v)}{N_v}$	u–web page, B(u)–set of pages that point to u, PR–ranking score, N_v–no of outgoing links of v, c– normalization factor
Modified Version	$PR(u) = (1-d) + d \sum_{v \in B(u)} \frac{PR(v)}{N_v}$	d – dampening factor
Weighted Page Rank	$WPR(u) = (1-d) + d \sum_{v \in B(u)} WPR(v) W_{(v,u)}^{in} W_{(v,u)}^{out}$ where $W_{(v,u)}^{in} = \frac{I_u}{\sum_{p \in R(v)} I_p}$ and $W_{(v,u)}^{out} = \frac{O_u}{\sum_{p \in R(v)} O_p}$	Iu, Ip – no of incoming links to page u, p. R(v) – Reference page list of v

Another efficient algorithm is HITS. Kleinberg [18] developed a WSM based algorithm called Hyperlink-Induced Topic Search (HITS) [5] which assumes that for every query given by the user, there is a set of authority pages that are relevant and popular focusing on the query and a set of hub pages that contain useful links to relevant pages/sites including links to many authorities. HITS assumes that if the author of page *p* provides a link to page *q*, then *p* confers some authority on page *q*.

The HITS algorithm considers the WWW as a directed graph *G (V, E)*, where *V* is a set of vertices representing pages and *E* is a set of edges that correspond to links. A directed edge *(p, q)* indicates a link from page *p* to page *q*. The search engine may not retrieve all relevant pages for the query; therefore the initial pages retrieved by the search engine are a good starting point to move further. But relying only on the initial pages does not guarantee that authority and hub pages are also retrieved efficiently. To remove this problem, HITS uses a proper method to find the relevant information regarding the user query.

Similarly the Jaccard coefficient is a measure for calculating the similarity (or diversity) between sets. The variation of the Jaccard coefficient used in this work is defined in Table.II. In probabilistic terms, it finds the maximum likelihood estimate of the ratio of the probability of finding a document, where words w_1 and w_2 co-occur over the probability of finding a document where either w_1 or w_2 occurs. If w_1 and w_2 are the same word, then the Jaccard coefficient is equal to 1 (is called absolute semantic similarity). If two words not co-occur in a document collection, then the Jaccard coefficient is 0. The Dice coefficient is related to the Jaccard

coefficient and is computed as mentioned in Table II. Again, the Dice coefficient is equal to 1 if w_1 and w_2 are identical, 0 otherwise.

Both Jaccard and Dice coefficients are compared in [1], and it is found that both of them do not differ significantly. Another measure is mutual information. If we assume that the number of documents indexed by the words w_1, w_2 are random variables X, Y , respectively, then the pointwise mutual information (MI) between X and Y measures the mutual dependence between the occurrence of words w_1 and w_2 [4]. The maximum likelihood estimate of MI is calculated as given in Table. II. Mutual information measures the sharing of information between variables *X* and *Y* share.

Table II. Jaccard and Dice coefficients

Scheme	*Equation*	*Description*
Jaccard coefficient	$J(w_1,w_2)=\frac{\lvert D\rvert w_1,w_2\rvert}{\lvert D\rvert w_1\rvert+\lvert D\rvert w_2\rvert-\lvert D\rvert w_1,w_2\rvert}$	$\{D\}$ - Set of all documents indexed by search engine $\lvert D\rvert$ - Number of documents in {D} w - A word or term $\{D\rvert w\}$ - Subset of {D}, documents indexed by w $\{D\rvert w1,w2\}$ - Subset of {D}, documents indexed by $w_1$and w_2 $\lvert D\rvert w\rvert$ - Number of documents in {D} indexed by w $\lvert D\rvert w1,w2\rvert$ - Number of documents in {D} indexed by w_1 and w_2
Dice coefficient	$C(w_1,w_2)=\frac{2\lvert D\rvert w_1,w_2\rvert}{\lvert D\rvert w_1\rvert+\lvert D\rvert w_2\rvert}$	
Mutual Information	$I(X,Y)=\log\frac{\frac{\lvert D\rvert w_1,w_2\rvert}{\lvert D\rvert}}{\frac{\lvert D\rvert w_1\rvert}{\lvert D\rvert}\frac{\lvert D\rvert w_2\rvert}{\lvert D\rvert}}$	

MI quantifies how the knowledge of one variable reduces the uncertainty about the other. For instance, if *X* and *Y* are independent, then knowing *X* does not give any information about *Y* and the mutual information is 0. For *X* = *Y*, the knowledge of *X* provides the value of *Y* with certainty and the mutual information is 1. Note that the number of relevant documents is normalized by the total number of documents indexed by the search engine, |*D*|, giving a maximum likelihood estimate of the probability of finding a document in the Web that contains this word.

Similarly, page – count based similarity metrics, known as Normalized Google Distance, which is described in the following section.

Google Similarity Distance

Motivated by Kolmogorov complexity, Cilibrasi and Vitanyi [39], [33] proposed a page-count-based similarity measure, called the Normalized Google Distance, defined as follows.

$$G(w_1,w_2)=\frac{\max\{A\}-\log\lvert D\rvert w_1,w_2\rvert}{\log\lvert D\rvert-\min\{A\}} \quad (5)$$

where $A=\{\log\lvert D\rvert w_1\rvert,\log\lvert D\rvert w_2\rvert\}$. As the semantic similarity between two words increases, the distance computed by the above equation decreases. Thus, this metric can be considered as a dissimilarity measure. Note that the metric is also unbounded, ranging from 0 to ∞. In [11], a variation of Normalized Google Distance (NGD) is proposed that defines a similarity measurement, named as "Google-based Semantic Relatedness":

$$G'(w_1,w_2)=e^{-2G(w_1,w_2)} \quad (6)$$

where $G(w_1,w_2)$ is computed according to (5). Note that the Google – based Semantic Relatedness is bounded taking values between 0 and 1. In [7], these methods are compared, and results show that the NGD outperforms than other similarity metrics.

This NGD can also be extended to compute sentence similarity [31]. First, using the NGD, the global and local dissimilarity measure between terms is defined. According to definition NGD the global dissimilarity measure between terms t_k and t_l also is defined by the formula:

$$NGD^{global}(t_k,t_l)=\frac{\max\left\{\log\left(f_k^{global}\right),\log\left(f_l^{global}\right)\right\}-\log\left(f_{kl}^{global}\right)}{\log N_{Google}-\min\left\{\log\left(f_k^{global}\right),\log\left(f_l^{global}\right)\right\}} \quad (7)$$

Using the formula (7), a global dissimilarity measure between sentences s_i and s_j as follows: as in

$$diss_{NGD}^{global}(s_i,s_j)=\frac{\Sigma t_k\in s_i\ \Sigma t_l\in s_j\ NGD^{global}(t_k,t_l)}{m_i m_j} \quad (8)$$

Similarly, a local dissimilarity measure between sentences s_i and s_j is defined as: as in

$$diss_{NGD}^{local}(s_i,s_j)=\frac{\Sigma t_k\in s_i\ \Sigma t_l\in s_j\ NGD^{local}(t_k,t_l)}{m_i m_j} \quad (9)$$

where, $$NGD^{local}(t_k,t_l)=\frac{\max\left\{\log\left(f_k^{local}\right),\log\left(f_l^{local}\right)\right\}-\log\left(f_{kl}^{local}\right)}{\log n-\min\left\{\log\left(f_k^{local}\right),\log\left(f_l^{local}\right)\right\}} \quad (10)$$

is the local dissimilarity measure between terms t_k and t_l, in which f_k^{local} denotes the number of sentences in a document D, containing the term t_k, and f_{kl}^{local} denotes the number of sentences containing both terms t_k and t_l. Thus the overall sentence dissimilarity is defined as a product of global and local dissimilarity measures: as in

$$diss_{NGD}(s_i,s_j)=diss_{NGD}^{local}(s_i,s_j)\cdot diss_{NGD}^{global}(s_i,s_j) \quad (11)$$

Finally NGD is compared to Euclidean distance, and it is found that NGD outperforms than the other. Similarly various approaches are used to compute the similarity. In [42], the proposed measure improves the Optimal Matching (OM) – based measure by using the earth mover's distance (EMD) to allow many-to-many matching between subtopics, thus benefiting the evaluation of document similarity based on subtopic structure.

The framework of the proposed EMD-based measure is similar to the OM-based measure, consisting of the following two steps: (1) Decompose documents into sets of subtopics; (2) Evaluate document similarity based on the subtopic sets. In the first step, different algorithms such as the TextTiling and the sentence clustering can be adopted to decompose documents.

In the second step, the proposed measure formalizes the comparison of two subtopic sets as the transportation problem. It adopts the earth mover's distance to solve the problem, while the previous OM-based measure formalizes it as the optimal matching problem. The OM–based measure adopts the Kuhn–Munkres algorithm to solve the problem. Sometimes a combinational approach is used for the efficient retrieval. In [43], both citation linkage information and content – based information retrieval weighting are used. It directly approximates citation links frequencies in datasets. Thus the citation score is used along with the text score, which is calculated by applying BM25 for weighting.

In contrast to the above work, ADSS [20] introduces a multi objective programming algorithm to compute the weights

and the Tabu Search to compute the optimal solution. The proposed approach can make use of both the distributed data about a new word and the strength of the semantic relatedness of its target class to the other candidate classes. This approach can acquire the results with higher precision.

V. RANKING

Measuring effectiveness of information retrieval (IR) systems is essential for research and development and for monitoring search quality in dynamic environments. In [29], methods for automatic ranking of retrieval systems are proposed. First, k systems to be fused are selected. System selection is based on best, normal, and bias criteria. The maximum number of systems that can be selected is the number of systems (n) in the test environment and; therefore, $k \leq n$. Then using the data fusion methods, the top b documents from each selected system are combined. '*S*' parameter is a threshold which is used to select the highest merging result and treated as Pseudorels. Finally, the performance of each retrieval system is evaluated and ranked using Pseudorels.

Three data fusion methods for determining the Pseudorels are the Rank Position, Borda Count, and Condorcet methods. The Rank position method in which the rank score of document i using the position information of this document in all of the systems *(j = 1. . .n)* is

$$r(d_i) = \frac{1}{\sum_j 1/position(d_{ij})} \quad (12)$$

In Borda count, the highest ranked individual (in an n-way rank) gets n count and each subsequent gets one count less (so the number two gets $n - 1$ and the number three gets $n - 2$ and so on). Document with high count treated as top rank document. In the Condorcet election method, voters rank the candidates in the order of preference. The vote counting procedure then takes into account each preference of each voter for one candidate over another. The Condorcet voting algorithm specifies the winner as the candidate, which beats each of the other candidates in a pair wise comparison. Even though it chooses the most relevant using the data fusion method over various ranking techniques, it depends on the results produced by the systems on which the data fusion is applied. Thus ranking in turn depends on the relatedness measure.

Similarly in [41] three types of ranking metrics are given (Table. III). These metrics are calculated mainly from usage and contextual information and do not require any explicit information from users. This metrics are evaluated in all the subjects from the same field and had similar teaching styles. With this homogeneity the results of the basic topical metric boosted because of the absence of noise in the data. But it is a question when it is applied to subjects in different field, where the topic diffusion may occur heavily.

Sometimes the relationships between documents and the information stored in a relational database may be uncertain, because they are from different data sources and the relationships are determined systematically using similarity match which causes uncertainty. In [3], provided a solution to rank the documents in a context where there exist uncertainty between the documents and the context. Here a context is a multi-attribute graph G, which can represent any information maintained in a relational database, where multi-attribute nodes represent tuples, and edges represent primary key and foreign key references among nodes.

The main issue is, finding top-k documents for an l-keyword query, $Q = \{w_1, w_2. . . w_l\}$, against graph G. The ranking is based on the following consideration. First, if a document node itself contains all the l-keywords, it should be ranked higher. If a document node, d_i, does not contain all the l-keywords, it is ranked based on d_i and its associated local context that together contain all the l-keywords. The selection of such a local context is done by considering all possibilities. Scoring functions given (Table. IV) with two components (the IR-styled score and the structural cost).

Table III. Ranking Algorithms

Scheme		*Equation*
Topic Relevance Ranking Metrics	Basic Topical Relevance Metric (BT)	$BT(o,q) = \sum_{i=1}^{NQ} distance(q,q_i) * selected(o,q_i)$ where $selected(o,q) = \begin{cases} 1, & if\ o\ clicked\ in\ q \\ 0, & otherwise \end{cases}$,
	Course-Similarity Topical Relevance Ranking (CST)	$CST(o,c) = \sum_{i=1}^{NC} SimRank(c,c_i) * present(o,c_i)$ where $SimRank(c_1,c_2) = \sum_{i=1}^{NO} present(o_i,c_1) * present(o_i,c_2)$ $present(o,c) = \begin{cases} 1, & if\ o \in c, \\ 0, & otherwise \end{cases}$
	Internal Topical Relevance Ranking (IT)	$IT(o) = authority(o) = \sum_{i=1}^{N} degree(c_i) \mid c_i\ includes\ o$
Personal Relevance Ranking Metrics *OptScore*	Basic Personal Relevance Ranking (BP)	$BP(o,u) = \sum_{i=1}^{NF} freq(u,f_i,val(o,f_i)) \mid f_i\ present\ in\ o$
	User-Similarity Personal Relevance Ranking (USP)	$UST(u,o) = \sum_{i=1}^{NU} SimRank(u,u_i) * has\ Reused(o,u_i)$ where $has\ Reused(o,u) = \begin{cases} 1, & if\ o\ used\ by\ u \\ 0, & otherwise \end{cases}$,
Situational Relevance Ranking Metrics	Basic Situational Relevance Ranking (BS)	$BS(o,t) = \frac{\sum_{i=1}^{M} tv_i * ov_i}{\sqrt{\sum_{i=1}^{M} tv_i^2 \cdot \sum_{i=1}^{M} ov_i^2}}$
	Context Similarity Situational Relevance Ranking (CSS)	$CCS(o,c) = \sum_{i=1}^{NF} freq(c,f_i,val(o,f_i)) \mid f_i\ present\ in\ o$ where $freq(c,f,v) = \frac{1}{N} \sum_{i=1}^{N} cont(o_i,f,v) \mid o_i\ included\ in\ c$

Table IV. Ranking Algorithms

Scheme	*Equation*
ERank	$ERank(d_i) = \sum_{T:doc(T)=d_i} ERank(T)$ where $ERank(T) = \sum_{I \in pwd(\chi), T \in I} rank_I(T) \cdot \Pr(I)$
tkp	$tkp(d_i) = \sum_{T:doc(T)=d_i} tkp(T)$ where $tkp(T) = \sum_{I \in pwd(\chi), T \in topk(I)} \Pr(I)$
EScore	$EScore(d_i) = \sum_{T:doc(T)=d_i} score(T,Q) \cdot prob(T)$
*Opt*Prob	Choose the score of a document as the score of the local context in the corresponding supporting set (x-tuple) with the largest probability and score, respectively.
OptScore	
DocOnly	Rank the documents based on the IR-score only, without a multi-attribute graph.

VI. OTHER ISSUES

Context capturing is a key issue in searching. In [16], a new pattern of context acquisition and inference in implementing the ubiquitous computing based-Information Systems (IS) along with the direction of internal context awareness research.

For inferring internal context in this domain, packet sniffing technique, is utilized. Packet-Based Context Acquisition Module (PBCAM) captures the packets on the way to the internet application, Word Vector Tool stems the words in the documents and transforming the words into the vector, and SVM determines the keyword of a document by mapping the word vectors.

In addition to context, ontologies are also most helpful to reduce the information overload. Ontologies offer an efficient way to reduce the amount of information overload by encoding the structure of the domain and facilitate easier access to the information for the user. The following section describes the various research issues in the field of IR with ontologies.

A. Ontology based Search

OntoSearch [45], a kind of "ontology Google", which can help users find ontologies on the Internet. It allows the user to perform keyword searches on certain types of “ontology” files, and to visually inspect the files to check their relevance. OntoSearch applies the Google engine to search for RDFs files related to the keywords and returns a list of relevant links (URLs) to the user. The user then chooses some of the returned RDFs files and displays their structure, and decides which of the files are relevant. Finally, the user selects the relevant RDFs files and saves them in a taxonomy library for future use. Once the relevant files are extracted then it is useful to perform more efficient search. But, it needs human intervention and not automatic.

Similarly in ORank [22], each document is annotated and a vector is created. The dimensions are weighted by statistical methods considering higher weights for phrases rather than single words. The ontology processor assigns weights to the relations in the reference ontology. First, the query phrases are extracted then weighted ontology is applied to expand these phrases with their related concepts. The query vector in each dimension is either corresponds to a query phrase or its expanded concepts. Finally, the rank of each document is calculated according to its relevancy to the expanded user query.

In [25], relational database query language (RDQL) query is generated from the given user query (Fig. 2). Explicitly the user has to select ontology classes and enter property values. Lists of instance tuples that satisfy the query are retrieved from the knowledge base. This step of the process is purely Boolean (i.e., based on an exact match), so that the returned instances must strictly hold all the conditions in the formal query. The documents are annotated with the instance returned by the previous step, and then it was returned but the number of conditions determine the ranking of the result set tuples. Explicit specification of ontology is a major drawback of this architecture compared to earlier architectures.

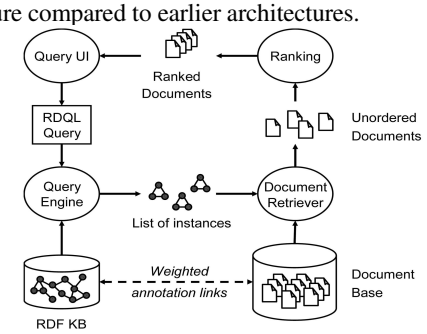

Figure 2. Pablo Castells et al. architecture.

To overcome this issue, in [26] takes advantage of the additional semantics (class hierarchies, precise and formalized relations) expressed by the ontology, that cannot be expressed using keywords or usage history (Fig. 3). Hence ranking can also be done based on the user interests.

For the efficient retrieval, not only context, thesaurus is also useful. In [21], a thesaurus-based semantic context-aware auto completion mechanism is proposed (Fig. 4). The context model provides a formal representation of the user, the environment and the access mechanism, enabling personalization and context – awareness. Thesaurus, which models semantic relationships (synonym, broader, narrower, related) among the specific concepts managed by the search system. Each thesaurus concept may have additional properties that are described in ontology.

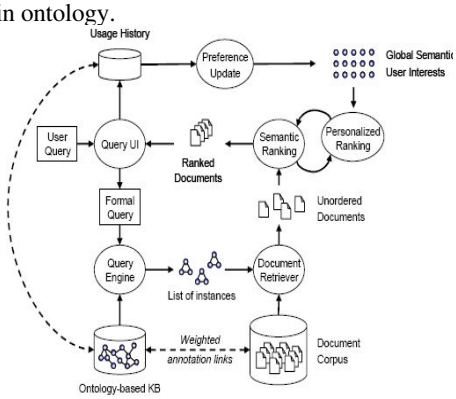

Figure 3. Pablo Castells et al. modified architecture.

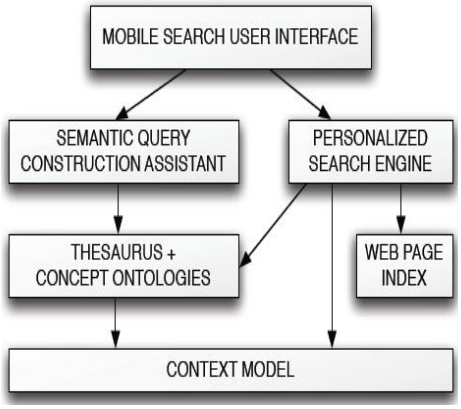

Figure 4. Semantic context – aware auto completion process.

A Semantic Query Construction Assistant intended to recommend best-suitable options in context to avoid users from typing, and thus personalizing the search process. A Personalized Search Engine which makes use of all information available (the query, selected thesaurus concept, ontology values) to find relevant results through a web page index. A separate link is to be established for each synonym in a thesaurus based tree. The more number of documents for synonyms leads to computation overhead.

Thus, searching with context, ontologies are useful, and efficient, rather than keyword search [35], [36].

B. Relation – based Search

The relationship between keywords is explicit only to the users, not to the search engines. Since the web page is a collection of keywords without having semantic relationships, it is too difficult to provide user required information. OntoLook [48], is a relation – based search engine (Fig. 5.). It retrieves the relations between the keywords of the given query, based on which a concept – relation graph is constructed. Appropriate relations from the arc of the concept – relation sub graph are selected, which forms property –

keyword candidate set. Using this candidate set, URL set is retrieved by the ontology database, which is redirected to the web page database to retrieve the web page information.

If the number of relationships between concepts may be large, the priority ranking of relationships will affect the returned pages. To overcome this issue, priority ranking technology and the page ranking technology are combined to make a "relation-based page rank" [8] (Fig. 6.). Using the page database, it builds the unordered result set including all pages containing keywords/ concepts in user query. The query and page sub graph is constructed, along with its page score for each page in the result set. Ordered result set is built by assigning pages to its relevance class. Finally page information is retrieved from the page database by means of ordered result set.

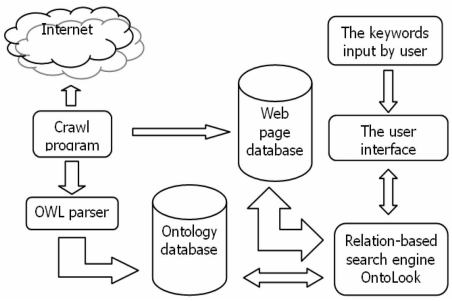

Figure 5. Architecture of OntoLook.

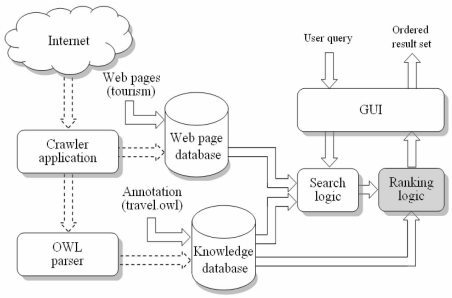

Figure 6. Fabrizio Lamberti et al. architecture.

Even though, relation between the keywords may not be expressed in the form of a graph, still the relations are used to make search efficient. Semsearch [47], [38], hides the complexity of semantic search from end users and making it easy to use and effective. First, the text search layer finds the semantic meanings of the keywords, specified in a user query. Then the semantic query layer translates the user query into formal queries, which is transferred to semantic data repositories. Finally the retrieved results are ranked and presented to the user.

VII. CONCLUSION

Most of the systems are not forcing the user to specify the query in a formal way, but almost they are missing in the extraction of context from the given user query. There are many directions in which the work can be continued. One direction is, the next level of auto completion of the query can be achieved only when the context is identified successfully, which is a major issue to be solved. Another direction is to improve the automatic context selection techniques by using a learning mechanism. When a user performs a context-based search, by providing a search term, potentially relevant contexts are listed, which in turn modified further by the user. Since the number of contexts of a given query can be quite large, which contexts to select are also needs to be investigated. As discussed earlier in the paper, the traditional methods used for representing a document, leads to more computational overhead as well as diffuses the topic. Thus, the document representation needs to be focused more by the researchers.

VIII. REFERENCES

[1] Benjamin Markines, Ciro Cattuto, Filippo Menczer, Domink Benz, Andreas Hotho, Gerd Stumme, Evaluating Similarity Measures for Emergent Semantics of Social Tagging, Proc. of World Wide Web Conference, Madrid, Spain, (2009).

[2] Campbell D.R, Culley S. J, McMahon C. A, Sellini F, An approach for the capture of context-dependent document relationships extracted from Bayesian analysis of users' interactions with information, Information Retrieval, 10 (2007) 115–141.

[3] Chang LJ, Yu J, Qin L, Context-Sensitive Document Ranking, Journal of Computer Science and Technology, 25(3) (2010) 444–457.

[4] Church.K, Hanks.P, Word Association Norms, Mutual Information, and Lexicography, Computational Linguistics, 16 (1) (1990) 22–29.

[5] Ding.C, He.X, Husbands.P, Zha.H, Simon.H, Link Analysis: Hubs and Authorities on the World, Technical report: 47847, (2001).

[6] Dino Isa, Lam Hong Lee, V.P. Kallimani, and R. RajKumar, Text Document Preprocessing with the Bayes Formula for Classification Using the Support Vector Machine, IEEE Transactions on Knowledge and Data Engineering, 20 (9) (2008).

[7] Elias Iosif, Alexandros Potamianos, Unsupervised Semantic Similarity Computation between Terms Using Web Documents, IEEE Transactions on Knowledge and Data Engineering, 22 (11) (2010).

[8] Fabrizio Lamberti, Andrea Sanna, Claudio Demartini, "A Relation-Based Page Rank Algorithm for Semantic Web Search Engines", IEEE Transactions on Knowledge and Data Engineering, 21 (1) (2009).

[9] Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E., Placing Search in Context: The Concept Revisited, Proc. of World Wide Web Conference, Hong Kong, China, (2001).

[10] Gae-won You, Seung-won Hwang, Search structures and algorithms for personalized ranking, Information Sciences, 178 (2008) 3925–3942.

[11] Gracia.J, Trillo.R, Espinoza.M, Mena.E, Querying the Web: A Multiontology Disambiguation Method, Proc. Int. Conf. Web Eng., California, USA, (2006), pp. 241–248.

[12] Hai Jin, Hanhua Chen, SemreX: Efficient search in a semantic overlay for literature retrieval, Future Generation Computer Systems, 24 (2008) 475–488.

[13] Hearst.M.A, TileBars: Visualization of Term Distribution Information in Full Text Information Access, In Proc. Of the ACM SIGCHI Conf. on Human Factor in Computing Systems, Denver, Colorado, USA, (1995), pp.59–66.

[14] Hearst, M.A., Pedersen, J.O., Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results, Proc. of SIGIR, Zurich, Switzerland, (1996).

[15] Information Explosion, http://news.netcraft.com/archives/web_server_survey.html .

[16] Hyeon Kang, Euiho Suh, Keedong Yoo, "Packet-based context aware system to determine information system user's context", Expert Systems with Applications 35 (2008) 286–300.

[17] Kaki, M., "Findex: Search Results Categories Help Users when Document Ranking Fails", ACM SIGCHI Conf. on

Human Factors in Computing Systems, Portland, OR, USA, (2005).

[18] Kleinberg J., Authorative Sources in a Hyperlinked Environment, Proc. of the 23rd annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, Melbourne, Australia, (1998).

[19] Kraft.R, Chang.C.C, Maghoul.F, Kumar.R, Searching with Context, Proc. of World Wide Web Conference, Edinburgh, Scotland, UK, (2006).

[20] Lixin Han, Linping Sun, Guihai Chen, Li Xie, ADSS: An approach to determining semantic similarity, Advances in Engineering Software, 3 (2006) 129–132.

[21] Mario Arias, Jose M. Cantera, Jesus Vegas, Context Based Personalization for Mobile Web Search, ACM, VLDB (2008).

[22] Mehrnoush Shamsfard, Azadeh Nematzadeh, Sarah Motiee, ORank: An Ontology Based System for Ranking Documents, International Journal of Computer Science, 1 (3) (2006).

[23] Neelam Duhan, Sharma.A.K, Komal Kumar Bhatia, Page Ranking Algorithms: A Survey, IEEE Int. Advance Computing Conf. (IACC) (Patiala, India, 6–7 March 2009).

[24] Osinski, S., Weiss, D., Conceptual Clustering using Lingo Algorithm: Evaluation on Open Directory Project Data, Advances in Soft Computing, Intelligent Information Processing and Web Mining, Proc. of the Int. IIS: IIPWM´04 Conf., (Zakopane, Poland, 2004), pp. 359–368.

[25] Pablo Castells, Miriam Fernandez, David Vallet, An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval, IEEE Transactions on Knowledge and Data Engineering, 19 (2) (2007).

[26] Pablo Castells, Miriam Fernández, David Vallet, Phivos Mylonas, Yannis Avrithis, Self-Tuning Personalized Information Retrieval in an Ontology-Based Framework, OTM Workshops, Agia Napa, Cyprus, (2005) pp. 977–986.

[27] Page.L, Brin.S, Motwani.R, Winograd.T, The Pagerank Citation Ranking: Bringing order to the Web, Technical report, Stanford Digital Libraries SIDL-WP (1999) pp.1999–0120.

[28] Paolo Coppola, Vincenzo Della Mea, Luca Di Gaspero, Davide Menegon, Danny Mischis, Stefano Mizzaro, Ivan Scagnetto, and Luca Vassena, The Context-Aware Browser, IEEE Intelligent Systems, (2010).

[29] Rabia Nuray, Fazli Can, Automatic ranking of information retrieval systems using data fusion, Information Processing and Management, 42 (2006) 595–614.

[30] Rajan.K, Ramalingam.V, Ganesan.M, Palanivel,.S, Palaniappan.B, Automatic classification of Tamil documents using vector space model and artificial neural network, Expert Systems with Applications, 36 (2009), 10914–10918.

[31] Ramiz M. Aliguliyev, A new sentence similarity measure and sentence based extractive technique for automatic text summarization, Expert Systems with Applications 36 (2009) 7764–7772.

[32] Rıdvan Saracoglu, Kemal Tutuncu, Novruz Allahverdi, A fuzzy clustering approach for finding similar documents using a novel similarity measure, Expert Systems with Applications, 33 (2007) 600–605.

[33] Rudi L. Cilibrasi, Paul M.B. Vitanyi, The Google Similarity Distance, IEEE Transactions on Knowledge and Data Engineering, 19 (3), (2007).

[34] Salton.G, Automatic Text Analysis, Science, 168 (3929) (1970), 335–342.

[35] Sridevi.U.K, Nagaveni .N, Ontology based Correlation Analysis in Information Retrieval, International Journal of Recent Trends in Engineering, 2 (1), (2009).

[36] Sridevi.U.K, Nagaveni .N, Ontology based Similarity Measure in Document Ranking, International Journal of Computer Applications (0975–8887), 1 (26), (2010).

[37] Tommy W.S. Chow, Haijun Zhang, M.K.M. Rahman, A new document representation using term frequency and vectorized graph connectionists with application to document retrieval, Expert Systems with Applications, 36 (2009), 12023–12035.

[38] Victoria Uren, Yuangui Lei, Enrico Motta, SemSearch: Refining Semantic Search, (2006).

[39] Vitanyi.P, Universal Similarity, Proc. of Information Theory Workshop Coding and Complexity (ITW), Rotorua, NewZealand (2005) pp. 238–243.

[40] Wei Song, Soon Cheol Park, Genetic algorithm for text clustering based on latent semantic indexing, Computers and Mathematics with Applications, 57 (2009), 1901–1907.

[41] Xavier Ochoa, Erik Duval, Relevance Ranking Metrics for Learning Objects, IEEE Transactions on Learning Technologies, 1 (1) (2008).

[42] Xiaojun Wan, A novel document similarity measure based on earth mover's distance, Information Sciences, 177 (2007) 3718–3730.

[43] Xiaoshi Yin, Jimmy Xiangji Huang, Zhoujun Li, Mining and Modeling Linkage Information from Citation Context for Improving Biomedical Literature Retrieval, Information Processing and Management, (2010) Article in Process.

[44] Yen-Liang Chen, Yu-Ting Chiu, "An IPC-based vector space model for patent retrieval", Information Processing and Management, (2010) Article in press.

[45] Yi Zhang, Wamberto Vasconcelos, Derek Sleeman, OntoSearch: An Ontology Search Engine, Research and Development in Intelligent Systems, XXI, Session la, (2005) 58–69.

[46] Yi-Chun Liao, A weight-based approach to information retrieval and relevance feedback, Expert Systems with Applications, 35 (2008) 254–261.

[47] Yuangui Lei, Victoria Uren, Enrico Motta, SemSearch: A Search Engine for the Semantic Web, Proc. of Managing Knowledge in a World of Networks (EKAW), Czech Republic, (2006), 238–245.

[48] Yufei Li, Yuan Wang, Xiaotao Huang, A Relation-Based Search Engine in Semantic Web, IEEE Transactions on Knowledge and Data Engineering, 19 (2) (2007).

[49] Zamir, O., Etzioni, O., Grouper: a Dynamic Clustering Interface to Web Search Results, Proc. of World Wide Web Conference, Toronto, Canada, (1999).

